

Interview Summary	Application No.	Applicant(s)	
	09/898,351	SUBRAHMANYAM ET AL.	
	Examiner Tuan A. Vu	Art Unit 2193	

All participants (applicant, applicant's representative, PTO personnel):

(1) Tuan A. Vu. (3) _____.

(2) John Wagner. (4) _____.

Date of Interview: 13 March 2007.

Type: a) Telephonic b) Video Conference
c) Personal [copy given to: 1) applicant 2) applicant's representative]

Exhibit shown or demonstration conducted: d) Yes e) No.
If Yes, brief description: _____

Claim(s) discussed: 1, 6, 11-12, 14-16.

Identification of prior art discussed: n/a.

Agreement with respect to the claims f) was reached. g) was not reached. h) N/A.

Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: See Continuation Sheet.

(A fuller description, if necessary, and a copy of the amendments which the examiner agreed would render the claims allowable, if available, must be attached. Also, where no copy of the amendments that would render the claims allowable is available, a summary thereof must be attached.)

THE FORMAL WRITTEN REPLY TO THE LAST OFFICE ACTION MUST INCLUDE THE SUBSTANCE OF THE INTERVIEW. (See MPEP Section 713.04). If a reply to the last Office action has already been filed, APPLICANT IS GIVEN A NON-EXTENDABLE PERIOD OF THE LONGER OF ONE MONTH OR THIRTY DAYS FROM THIS INTERVIEW DATE, OR THE MAILING DATE OF THIS INTERVIEW SUMMARY FORM, WHICHEVER IS LATER, TO FILE A STATEMENT OF THE SUBSTANCE OF THE INTERVIEW. See Summary of Record of Interview requirements on reverse side or on attached sheet.



Examiner Note: You must sign this form unless it is an
Attachment to a signed Office action.

Examiner's signature, if required

Summary of Record of Interview Requirements

Manual of Patent Examining Procedure (MPEP), Section 713.04, Substance of Interview Must be Made of Record

A complete written statement as to the substance of any face-to-face, video conference, or telephone interview with regard to an application must be made of record in the application whether or not an agreement with the examiner was reached at the interview.

Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews Paragraph (b)

In every instance where reconsideration is requested in view of an interview with an examiner, a complete written statement of the reasons presented at the interview as warranting favorable action must be filed by the applicant. An interview does not remove the necessity for reply to Office action as specified in §§ 1.111, 1.135. (35 U.S.C. 132)

37 CFR §1.2 Business to be transacted in writing.

All business with the Patent or Trademark Office should be transacted in writing. The personal attendance of applicants or their attorneys or agents at the Patent and Trademark Office is unnecessary. The action of the Patent and Trademark Office will be based exclusively on the written record in the Office. No attention will be paid to any alleged oral promise, stipulation, or understanding in relation to which there is disagreement or doubt.

The action of the Patent and Trademark Office cannot be based exclusively on the written record in the Office if that record is itself incomplete through the failure to record the substance of interviews.

It is the responsibility of the applicant or the attorney or agent to make the substance of an interview of record in the application file, unless the examiner indicates he or she will do so. It is the examiner's responsibility to see that such a record is made and to correct material inaccuracies which bear directly on the question of patentability.

Examiners must complete an Interview Summary Form for each interview held where a matter of substance has been discussed during the interview by checking the appropriate boxes and filling in the blanks. Discussions regarding only procedural matters, directed solely to restriction requirements for which interview recordation is otherwise provided for in Section 812.01 of the Manual of Patent Examining Procedure, or pointing out typographical errors or unreadable script in Office actions or the like, are excluded from the interview recordation procedures below. Where the substance of an interview is completely recorded in an Examiners Amendment, no separate Interview Summary Record is required.

The Interview Summary Form shall be given an appropriate Paper No., placed in the right hand portion of the file, and listed on the "Contents" section of the file wrapper. In a personal interview, a duplicate of the Form is given to the applicant (or attorney or agent) at the conclusion of the interview. In the case of a telephone or video-conference interview, the copy is mailed to the applicant's correspondence address either with or prior to the next official communication. If additional correspondence from the examiner is not likely before an allowance or if other circumstances dictate, the Form should be mailed promptly after the interview rather than with the next official communication.

The Form provides for recordation of the following information:

- Application Number (Series Code and Serial Number)
- Name of applicant
- Name of examiner
- Date of interview
- Type of interview (telephonic, video-conference, or personal)
- Name of participant(s) (applicant, attorney or agent, examiner, other PTO personnel, etc.)
- An indication whether or not an exhibit was shown or a demonstration conducted
- An identification of the specific prior art discussed
- An indication whether an agreement was reached and if so, a description of the general nature of the agreement (may be by attachment of a copy of amendments or claims agreed as being allowable). Note: Agreement as to allowability is tentative and does not restrict further action by the examiner to the contrary.
- The signature of the examiner who conducted the interview (if Form is not an attachment to a signed Office action)

It is desirable that the examiner orally remind the applicant of his or her obligation to record the substance of the interview of each case. It should be noted, however, that the Interview Summary Form will not normally be considered a complete and proper recordation of the interview unless it includes, or is supplemented by the applicant or the examiner to include, all of the applicable items required below concerning the substance of the interview.

A complete and proper recordation of the substance of any interview should include at least the following applicable items:

- 1) A brief description of the nature of any exhibit shown or any demonstration conducted,
- 2) an identification of the claims discussed,
- 3) an identification of the specific prior art discussed,
- 4) an identification of the principal proposed amendments of a substantive nature discussed, unless these are already described on the Interview Summary Form completed by the Examiner,
- 5) a brief identification of the general thrust of the principal arguments presented to the examiner,
(The identification of arguments need not be lengthy or elaborate. A verbatim or highly detailed description of the arguments is not required. The identification of the arguments is sufficient if the general nature or thrust of the principal arguments made to the examiner can be understood in the context of the application file. Of course, the applicant may desire to emphasize and fully describe those arguments which he or she feels were or might be persuasive to the examiner.)
- 6) a general indication of any other pertinent matters discussed, and
- 7) if appropriate, the general results or outcome of the interview unless already described in the Interview Summary Form completed by the examiner.

Examiners are expected to carefully review the applicant's record of the substance of an interview. If the record is not complete and accurate, the examiner will give the applicant an extendable one month time period to correct the record.

Examiner to Check for Accuracy

If the claims are allowable for other reasons of record, the examiner should send a letter setting forth the examiner's version of the statement attributed to him or her. If the record is complete and accurate, the examiner should place the indication, "Interview Record OK" on the paper recording the substance of the interview along with the date and the examiner's initials.

Continuation of Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: The representative was contacted in order to effect some addition to the claims in regard to the use of the block frequency counter. That is, the Examiner indicated that there must be a rationale behind using this counter as a cached counter and its being stored outside of the runtime cache when the corresponding block of code is evicted from the code cache. It is then believed that the claim(s) should put forth the use of said counter while in cache AND outside of cache in terms of its subsequent support for tracking the evicted code when this code is reiterated back into cache. The Examiner also mentioned about putting the executable aspect of the blocks of code (under analysis) into the claim preamble to obviate prior art teaching. The representative agreed on following the above suggestions, so that when properly amended and free of minor deficiencies, the claims would be attached to the Examiner's Amendments, by way of which the amended subject matter therein would be in condition of allowance, which has been the very purpose of the interview.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:)
Subrahmanyam) Examiner: Vu, T.
Serial No.: 09/898,351) Art Unit: 2193
Filed: 07/03/2001)
For: A System and Method to)
Decrease Program Analysis)
Overhead)

Examiner T. Vu,
United States Patent and Trademark Office
Alexandria, Virginia 22313

AMENDMENTS TO CLAIMS PER AGREEMENT WITH EXAMINER

ON MARCH 15, 2007

Dear Examiner Vu:

In response to our telephonic interview on March 13th, your email of March 15th, and our conversations on March 16th, 2007, regarding our proposed claim amendments, we submit the following amended claim set for this case. Applicant respectfully requests the Examiner to enter the following amendments for allowance of this case.

Amendments to the Claims

1. (currently amended) A method to analyze a computer program that includes a plurality of executable blocks of code, the method comprising:
 - receiving a block of code to a code cache;
 - using a code block frequency counter for tracking each time said block of code is executed on said code cache, wherein additional code for incrementing said code block frequency counter is dynamically added to said block of code as said computer program is executed;
 - maintaining a counter cache for storing each said code block frequency counter of said block of code while said block of code is stored on said code cache, wherein said counter cache is distinct from said code cache, such that when said block of code is executed said code block frequency counter is readily available for tracking said execution; and
 - maintaining a storage area for storing each said code block frequency counter of said block of code previously executed on said code cache, said storage area distinct from said counter cache and said code cache, said code block frequency counter being stored in said storage area after said block of code is evicted from said code cache, said block of code being evicted to make room for another block of code in said code cache, said code block frequency counter being stored for future access in a location which is distinct from said code cache and said counter cache; and
in the event that said block of code is again received into said code cache for execution subsequent to being evicted, copying said code block frequency counter from

said storage area to said counter cache to enable continuation of said tracking each time said block of code is executed on said code cache.

2. (previously presented) The method of Claim 1, further comprising the step of:

identifying when said code cache is full.

3. (canceled)

4. (previously presented) The method of Claim 2, further comprising:

determining which said code block frequency counter of said block of code stored on said counter cache is least recently executed;

evicting said least recently executed block of code, related to said code block frequency counter, from said code cache; and

copying said code block frequency counter of said least recently executed block of code from said counter cache to said storage area when said least recently executed block of code related to said code block frequency counter is evicted from said code cache.

5. (previously presented) The method of Claim 1, wherein said receiving a block of code to a code cache further comprises:

checking said storage area to determine if said block of code is being executed for other than the first time;

loading said code block frequency counter associated with said block of code being executed for other than the first time, from said storage area into said counter cache; and

updating said code block frequency counter associated with said block of code being executed for other than the first time.

6. (currently amended) A computer implemented system having a computer for analyzing a computer program that includes a plurality of blocks of code, comprising:

means for executing said computer program;

means for maintaining a code cache for storing at least one of a plurality of blocks of code derived from said computer program;

means for counting each time one of said plurality of blocks of code is executed, wherein additional code for incrementing said code counting means is dynamically added to said one of said plurality of blocks block of code as said computer program is executed;

means for maintaining a counter cache for storing said counting means of said plurality of blocks of code that are most recently executed, wherein said counter cache is distinct from said code cache, such that when said one of said plurality of blocks of code is executed said counting means is readily available for tracking said execution;

and

means for maintaining a storage area for storing said counting means of said plurality of blocks of code that are not most recently executed, said storage area distinct from said counter cache and said code cache, said code counting means being stored

in said storage area after said block of code related to said counting means is evicted from said code cache, said block of code being evicted to make room for another block of code in said code cache, said counting means being stored for future access in a location which is distinct from said code cache and said counter cache, such that, subsequent to being evicted, if said block of code related to said counting means is again received into said code cache, said code counting means may be copied from said storage area to said counter cache to enable continuation of said tracking each time said block of code related to said counting means is executed within said code cache.

7. (previously presented) The system of Claim 6, further comprising:

means for identifying when said code cache is full.

8. (previously presented) The system of Claim 7, further comprising:

means for copying said counting means of said plurality of blocks of code from said code cache to said storage area when said code cache is full.

9. (previously presented) The system of Claim 8, wherein said identifying means further comprises:

means for determining which said counting means of said plurality of blocks of code in said code cache is least recently executed;

means for evicting said least recently executed block of code, related to said counter, from said code cache; and

means for copying said counting means, related to said least recently executed block of code, from said code cache to said storage area when said code cache is full.

10. (previously presented) The system of Claim 8, further comprising:

means for checking a code cache to determine if a block of code is being executed for other than the first time; and
means for loading said counting means associated with said block of code being executed for other than the first time, into said counter cache.

11. (currently amended) A computer readable storage medium having computer-readable program code embodied therein for causing a computer system to perform a method for analyzing a computer program that includes a plurality of executable blocks of code comprising:

receiving a block of code to a code cache;
utilizing a code block frequency counter for tracking each time said block of code is executed on said code cache, wherein additional code for incrementing said code block frequency counter is dynamically added to said block of code as said computer program is executed;

maintaining a counter cache for storing each said code block frequency counter of said block of code while said block of code is stored on said code cache, wherein said counter cache is distinct from said code cache, such that when said block of code is executed said code block frequency counter is readily available for tracking said execution; and

maintaining a storage area for storing each said code block frequency counter of said block of code previously executed on said code cache, said storage area distinct from said code cache and said counter cache, said code block frequency counter being stored in said storage area after said block of code is evicted from said code cache, said block of code being evicted to make room for another block of code in said code cache, said code block frequency counter being stored for future access in a location which is distinct from said code cache and said counter cache; and
in the event that said block of code is again received into said code cache for execution subsequent to being evicted, copying said code block frequency counter from said storage area to said counter cache to enable continuation of said tracking each time said block of code is executed on said code cache.

12. (previously presented) The computer readable storage medium of Claim 11, further comprising:

identifying when said code cache is full.

13. (canceled)

14. (previously presented) The computer readable storage medium of Claim 12, further comprises:

determining which said code block frequency counter of said block of code in said counter cache is least recently executed;

evicting said least recently executed block of code, related to said code block frequency counter, from said code cache; and

copying said code block frequency counter of said least recently executed block of code from said counter cache to said storage area when said least recently executed block of code related to said code block frequency counter is evicted from said code cache.

15. (currently amended) The computer readable storage medium of Claim 12 Claim 13, wherein said receiving a block of code to a code cache further comprises:

 checking said storage area to determine if said block of code is being executed for other than the first time;

 loading said code block frequency counter associated with said block of code being executed for other than the first time, from said storage area into said counter cache; and

 updating said code block frequency counter associated with said block of code being executed for other than the first time.

16. (currently amended) A system for analyzing a computer program that includes a plurality of executable blocks of code, the system comprising:

a computer system having a computer configured for executing a block of code,
said computer system comprising a cache memory;

 a code block frequency counter that tracks each time a specific block of code is executed by a code cache, wherein additional code for incrementing said code block

frequency counter is dynamically added to said block of code as said computer program is executed;

a counter cache for storing said code block frequency counter of a specific block of code while said specific block of code is stored on said code cache, such that when said block of code is executed said code block frequency counter is readily available for tracking said execution, wherein said counter cache is distinct from said code cache; and

a storage area for storing said code block frequency counter of a specific block of code previously executed on said code cache, said storage area distinct from said counter cache and said code cache, said code block frequency counter being stored in said storage area after said specific block of code is evicted from said code cache to make room for another block of code in said code cache, said code block frequency counter being stored for future access in a location which is distinct from said code cache and said counter cache such that, in the event that said block of code is again received into said code cache for execution, subsequent to being evicted, said code block frequency counter may be copied from said storage area to said counter cache to enable continuation of said tracking each time said block of code is executed on said code cache.

17. (previously presented) The system of Claim 16, further comprising:

logic that identifies when said code cache is full.

18. (canceled)

19. (previously presented) The system of Claim 17, wherein said logic determines which said code block frequency counter of said specific block of code stored on said counter cache is least recently executed, evicting said least recently executed block of code related to said code block frequency counter from said code cache, and copies said code block frequency counter of said specific block of code from said counter cache to said storage area when said least recently executed specific block of code is evicted from said code cache.

20. (previously presented) The system of Claim 17, wherein said logic checks said storage area to determine if said specific block of code is being executed for other than the first time, and loads said code block frequency counter associated with said specific block of code being executed for other than the first time, from said storage area into said counter cache, and updating said code block frequency counter associated with said specific block of code being executed for other than the first time.

Conclusion

In light of the above amendments made per agreement with Examiner Vu between March 13th and March 16th, 2007, Applicants respectfully request allowance of Claims 1-2, 4-12, 14-17 and 19-20. The Examiner is authorized to effect execution of the above amendments.

The Examiner is invited to contact Applicant's undersigned representative if the Examiner believes such action would expedite resolution of the present application.

Respectfully submitted,

WAGNER, MURABITO & HAO LLP

Dated: March 15, 2007

//signed//

John P. Wagner
Reg. No. 35,398

WAGNER, MURABITO & HAO LLP
Westridge Business Park
123 Westridge Drive
Watsonville, CA 95076
San Jose, CA 95113

Phone: (408) 938-9060
Facsimile: (831) 763-2895

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:)
Subrahmanyam) Examiner: Vu, T.
Serial No.: 09/898,351) Art Unit: 2193
Filed: 07/03/2001)
For: A System and Method to)
Decrease Program Analysis)
Overhead)

)

Examiner T. Vu,
United States Patent and Trademark Office
Alexandria, Virginia 22313

AMENDMENTS TO CLAIMS PER AGREEMENT WITH EXAMINER

ON MARCH 29, 2007

Dear Examiner Vu:

In response to our telephonic interview on March 13th, your email of March 15th, our conversations on March 16th, 2007, and our conversation on March 29, 2007, regarding our proposed claim amendments, we submit and agree to the following amended claim set for this case. Applicant respectfully requests the Examiner to enter the following amendments for allowance of this case.

Amendments to the Claims

1. (currently amended) A method to analyze a computer program that includes a plurality of executable blocks of code, the method comprising:
 - receiving a block of code to a code cache;
 - using a code block frequency counter for tracking each time said block of code is executed on said code cache, wherein additional code for incrementing said code block frequency counter is dynamically added to said block of code as said computer program is executed;
 - maintaining a counter cache for storing each said code block frequency counter of said block of code while said block of code is stored on said code cache, wherein said counter cache is distinct from said code cache, such that when said block of code is executed said code block frequency counter is readily available for tracking said execution; and
 - maintaining a storage area for storing each said code block frequency counter of said block of code previously executed on said code cache, said storage area distinct from said counter cache and said code cache, said code block frequency counter being stored in said storage area after said block of code is evicted from said code cache, said block of code being evicted to make room for another block of code in said code cache, said code block frequency counter being stored for future access in a location which is distinct from said code cache and said counter cache; and
in the event that said block of code is again received into said code cache for execution subsequent to being evicted, copying said code block frequency counter from

said storage area to said counter cache to enable continuation of said tracking each time said block of code is executed on said code cache.

2. (previously presented) The method of Claim 1, further comprising the step of:

identifying when said code cache is full.

3. (canceled)

4. (previously presented) The method of Claim 2, further comprising:

determining which said code block frequency counter of said block of code stored on said counter cache is least recently executed;

evicting said least recently executed block of code, related to said code block frequency counter, from said code cache; and

copying said code block frequency counter of said least recently executed block of code from said counter cache to said storage area when said least recently executed block of code related to said code block frequency counter is evicted from said code cache.

5. (previously presented) The method of Claim 1, wherein said receiving a block of code to a code cache further comprises:

checking said storage area to determine if said block of code is being executed for other than the first time;

loading said code block frequency counter associated with said block of code being executed for other than the first time, from said storage area into said counter cache; and

updating said code block frequency counter associated with said block of code being executed for other than the first time.

6. (currently amended) A computer implemented system having a computer for analyzing a computer program that includes a plurality of blocks of code, comprising:

means for executing said computer program;

means for maintaining a code cache for storing at least one of a plurality of blocks of code derived from said computer program;

means for counting each time one of said plurality of blocks of code is executed, wherein additional code for incrementing said code counting means is dynamically added to said one of said plurality of blocks block of code as said computer program is executed;

means for maintaining a counter cache for storing said counting means of said plurality of blocks of code that are most recently executed, wherein said counter cache is distinct from said code cache, such that when said one of said plurality of blocks of code is executed said counting means is readily available for tracking said execution; and

means for maintaining a storage area for storing said counting means of said plurality of blocks of code that are not most recently executed, said storage area distinct from said counter cache and said code cache, said code counting means being stored

in said storage area after said block of code related to said counting means is evicted from said code cache, said block of code being evicted to make room for another block of code in said code cache, said counting means being stored for future access in a location which is distinct from said code cache and said counter cache, such that, subsequent to being evicted, if said block of code related to said counting means is again received into said code cache, said code counting means may be copied from said storage area to said counter cache to enable continuation of said tracking each time said block of code related to said counting means is executed within said code cache.

7. (previously presented) The system of Claim 6, further comprising:

means for identifying when said code cache is full.

8. (previously presented) The system of Claim 7, further comprising:

means for copying said counting means of said plurality of blocks of code from said code cache to said storage area when said code cache is full.

9. (previously presented) The system of Claim 8, wherein said identifying means further comprises:

means for determining which said counting means of said plurality of blocks of code in said code cache is least recently executed;

means for evicting said least recently executed block of code, related to said counter, from said code cache; and

means for copying said counting means, related to said least recently executed block of code, from said code cache to said storage area when said code cache is full.

10. (previously presented) The system of Claim 8, further comprising:

means for checking a code cache to determine if a block of code is being executed for other than the first time; and

means for loading said counting means associated with said block of code being executed for other than the first time, into said counter cache.

11. (currently amended) A computer readable storage medium having computer-readable program code embodied therein for causing a computer system to perform a method for analyzing a computer program that includes a plurality of executable blocks of code comprising:

receiving a block of code to a code cache;

utilizing a code block frequency counter for tracking each time said block of code is executed on said code cache, wherein additional code for incrementing said code block frequency counter is dynamically added to said block of code as said computer program is executed;

maintaining a counter cache for storing each said code block frequency counter of said block of code while said block of code is stored on said code cache, wherein said counter cache is distinct from said code cache, such that when said block of code is executed said code block frequency counter is readily available for tracking said execution; and

maintaining a storage area for storing each said code block frequency counter of said block of code previously executed on said code cache, said storage area distinct from said code cache and said counter cache, said code block frequency counter being stored in said storage area after said block of code is evicted from said code cache, said block of code being evicted to make room for another block of code in said code cache, said code block frequency counter being stored for future access in a location which is distinct from said code cache and said counter cache; and in the event that said block of code is again received into said code cache for execution subsequent to being evicted, copying said code block frequency counter from said storage area to said counter cache to enable continuation of said tracking each time said block of code is executed on said code cache.

12. (previously presented) The computer readable storage medium of Claim 11, further comprising:

identifying when said code cache is full.

13. (canceled)

14. (previously presented) The computer readable storage medium of Claim 12, further comprises:

determining which said code block frequency counter of said block of code in said counter cache is least recently executed;

evicting said least recently executed block of code, related to said code block frequency counter, from said code cache; and

copying said code block frequency counter of said least recently executed block of code from said counter cache to said storage area when said least recently executed block of code related to said code block frequency counter is evicted from said code cache.

15. (currently amended) The computer readable storage medium of Claim 12 Claim 13, wherein said receiving a block of code to a code cache further comprises:

 checking said storage area to determine if said block of code is being executed for other than the first time;

 loading said code block frequency counter associated with said block of code being executed for other than the first time, from said storage area into said counter cache; and

 updating said code block frequency counter associated with said block of code being executed for other than the first time.

16. (currently amended) A system for analyzing a computer program that includes a plurality of executable blocks of code, the system comprising:

a computer system having a computer configured for executing a block of code,
said computer system comprising a cache memory;

 a code block frequency counter that tracks each time a specific block of code is executed by a code cache, wherein additional code for incrementing said code block

frequency counter is dynamically added to said block of code as said computer program is executed;

a counter cache for storing said code block frequency counter of a specific block of code while said specific block of code is stored on said code cache, such that when said block of code is executed said code block frequency counter is readily available for tracking said execution, wherein said counter cache is distinct from said code cache; and

a storage area for storing said code block frequency counter of a specific block of code previously executed on said code cache, said storage area distinct from said counter cache and said code cache, said code block frequency counter being stored in said storage area after said specific block of code is evicted from said code cache to make room for another block of code in said code cache, said code block frequency counter being stored for future access in a location which is distinct from said code cache and said counter cache such that, in the event that said block of code is again received into said code cache for execution, subsequent to being evicted, said code block frequency counter may be copied from said storage area to said counter cache to enable continuation of said tracking each time said block of code is executed on said code cache.

17. (previously presented) The system of Claim 16, further comprising:

logic that identifies when said code cache is full.

18. (canceled)

19. (previously presented) The system of Claim 17, wherein said logic determines which said code block frequency counter of said specific block of code stored on said counter cache is least recently executed, evicting said least recently executed block of code related to said code block frequency counter from said code cache, and copies said code block frequency counter of said specific block of code from said counter cache to said storage area when said least recently executed specific block of code is evicted from said code cache.

20. (previously presented) The system of Claim 17, wherein said logic checks said storage area to determine if said specific block of code is being executed for other than the first time, and loads said code block frequency counter associated with said specific block of code being executed for other than the first time, from said storage area into said counter cache, and updating said code block frequency counter associated with said specific block of code being executed for other than the first time.

Conclusion

In light of the above amendments made per agreement with Examiner Vu between March 13th and March 16th, 2007, Applicants respectfully request allowance of Claims 1-2, 4-12, 14-17 and 19-20. The Examiner is authorized to effect execution of the above amendments.

The Examiner is invited to contact Applicant's undersigned representative if the Examiner believes such action would expedite resolution of the present application.

Respectfully submitted,

WAGNER, MURABITO & HAO LLP

Dated: March 29, 2007

//signed//

John P. Wagner
Reg. No. 35,398

WAGNER, MURABITO & HAO LLP
Westridge Business Park
123 Westridge Drive
Watsonville, CA 95076
San Jose, CA 95113

Phone: (408) 938-9060
Facsimile: (831) 763-2895